The Thinnest Path Problem for Secure Communications: A Directed Hypergraph Approach

Jianhang Gao†, Qing Zhao†, Ananthram Swami§
†University of California, Davis, CA 95616, {jhgao,qzhao}@ucdavis.edu
§Army Research Laboratory, Adelphi, MD 20783, a.swami@ieee.org

***Abstract*— We formulate and study the thinnest path problem in wireless ad hoc networks. The objective is to find a path from the source to the destination that results in the minimum number of nodes overhearing the message by carefully choosing the relaying nodes and their corresponding transmission power. We adopt a directed hypergraph model of the problem and establish the NP-completeness of the problem in 2-D networks. We then develop a polynomial-time approximation algorithm that offers a $\sqrt{\frac{n}{2}}$ approximation ratio for general directed hypergraphs (which can model non-isomorphic signal propagation in space) and constant approximation ratio for disk hypergraphs (which result from isomorphic signal propagation). We also consider the thinnest path problem in 1-D networks and 1-D networks embedded in 2-D field of eavesdroppers with arbitrary unknown locations (the so-called 1.5-D networks). We propose a linear-complexity algorithm based on nested backward induction that obtains the optimal solution to both 1-D and 1.5-D networks. In particular, no algorithm, even with the complete knowledge of the locations of the eavesdroppers, can obtain a thinner path than the proposed algorithm which does not require the knowledge of eavesdropper locations.**

I. Introduction

A. The thinnest path problem

We consider the *thinnest path* problem in wireless ad hoc networks. For a given source and a destination, the thinnest path problem asks for a path from the source to the destination that results in the minimum number of nodes overhearing the message. Such a path is achieved by carefully choosing a sequence of relaying nodes and their corresponding transmission power.

At the first glance, one may wonder whether the thinnest path problem is simply a shortest path problem with the weight of each hop (i.e., each edge) given by the number of nodes who can hear the message in this hop. Realizing that a node may be within the transmission range of multiple relaying nodes and should not be counted multiple times in the total weight (referred to as the width) of the resulting path, we see that the thinnest path problem does not have a simple cost function that is summable over edges. But rather, the width of a path is given by the cardinality of the *union* of all receiving nodes in each hop, which is a highly nonlinear function of the weight of each hop. One may then wonder whether we can redefine the weight of each hop as the number of nodes that hear the message for the first time. Such a definition of edge weight indeed leads to a summable cost function. Unfortunately, in this case, the edge weight cannot be predetermined until the thinnest path from the source to the edge in question has already been established. The thinnest path problem is thus much more complex than the shortest path problem. Indeed, we show in this paper that the thinnest path problem in a 2-D network is NP-Complete, which is in sharp contrast with the polynomial nature of the shortest path problem.

Another aspect that complicates the problem is the design choice of the transmission power at each node (within a maximum value that may vary across nodes). In this case, the network cannot be modelled as a simple graph in which the neighbors of each node are prefixed. In this paper, we adopt the *directed hypergraph* model which easily captures the choice of different neighbor sets at each node. While a graph is given by a vertex set V and an edge set E consisting of cardinality-2 subsets of V, a hypergraph [1] is free of the constraint on the cardinality of an edge. Specifically, any non-empty subset of V can be an element (referred to as a hyperedge) of the edge set E. Hypergraphs can thus capture group behaviors and higher-dimensional relationships in complex networks that are more than a simple union of pairwise relationships. In a directed hypergraph [5], each hyperedge is directed going from a single source vertex to a non-empty set of destination

[0]This work was supported by the Army Research Laboratory Network Science CTA under Cooperative Agreement W911NF-09-2-0053.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
OCT 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	
The Thinnest Path Problem for Secure Communications: A Directed Hypergraph Approach	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Army Research Laboratory,Adelphi,MD,20783**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

in Proc. of the 50th Allerton Conference on Communications, Control, and Computing, October, 2012.

14. ABSTRACT

We formulate and study the thinnest path problem in wireless ad hoc networks. The objective is to find a path from the source to the destination that results in the minimum number of nodes overhearing the message by carefully choosing the relaying nodes and their corresponding transmission power. We adopt a directed hypergraph model of the problem and establish the NP-completeness of the problem in 2-D networks. We then develop a polynomial-time approximation algorithm that offers a n 2 approximation ratio for general directed hypergraphs (which can model non-isomorphic signal propagation in space) and constant approximation ratio for disk hypergraphs (which result from isomorphic signal propagation). We also consider the thinnest path problem in 1-D networks and 1-D networks embedded in 2-D field of eavesdroppers with arbitrary unknown locations (the so-called 1.5-D networks). We propose a linear-complexity algorithm based on nested backward induction that obtains the optimal solution to both 1-D and 1.5-D networks. In particular, no algorithm, even with the complete knowledge of the locations of the eavesdroppers, can obtain a thinner path than the proposed algorithm which does not require the knowledge of eavesdropper locations.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**6**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

vertices (see Fig. 1-(a) for a directed hypergraph with 3 vertices and 3 directed hyperedges). The directed hypergraph model of the thinnest path problem with power control is thus readily seen: rooted at each node (i.e., the source vertex) are multiple directed hyperedges, each corresponding to a distinct neighbor set feasible under the maximum transmission power of this node. We then arrive at a concrete formulation of the problem: the thinnest path problem with power control is to find a minimum-width hyperpath from the source to the destination where the width of a hyperpath is given by the cardinality of the union of the hyperedges on this hyperpath.

Based on the directed hypergraph formulation of the problem, we first show that the thinnest path problem in 2-D networks is NP-complete by reducing the problem to the minimum dominating set problem in graphs, a classic NP-complete problem. We further show that even with a fixed transmission power at each node (in this case, the resulting hypergraph degenerates to a standard graph), the thinnest path problem is NP-Complete. We then propose a polynomial-time approximation algorithm that offers a $\sqrt{\frac{n}{2}}$ approximation ratio(n is the number of nodes in the network) for general directed hypergraphs (which can model non-isomorphic signal propagation in space) and constant approximation ratios for disk hypergraphs (which result from isomorphic signal propagation).

We then consider the thinnest path problem in 1-D and 1.5-D networks. In a 1-D network, nodes are located on a line with arbitrary spacing and arbitrary maximum transmission power. The problem is non-trivial due to the arbitrary maximum power at each node which may render the use of backward links (relaying toward the opposite direction of the destination) necessary. Finding the thinnest path is thus equivalent to minimizing the number of nodes to the left of the source (assuming the destination is to the right of the source) that can overhear the message, and our algorithm is based on a nested backward induction starting at the destination. Referred to as NBI (nested backward induction), this algorithm has $O(n)$ time complexity. Since the size of the input date is $O(n)$, the proposed algorithm is order optimal. For the 1.5-dimension problem, we have a 1-D network embedded in a 2-D field of eavesdroppers with arbitrary unknown locations. Eavesdroppers, of course, will not be considered for relaying messages. The objective is to find a path with the minimum overhearing cost incurred at both the in-network nodes and the eavesdroppers (which may have a higher overhearing cost). We show that the NBI algorithm we developed for 1-D networks directly applies to the 1.5-D problem. More specifically, no algorithm, even with the complete knowledge of the locations of the eavesdroppers, can obtain a thinner path than the NBI algorithm which does not require the knowledge of eavesdropper locations. This result is obtained by establishing a strong property of the 1-D thinnest path algorithm: the area covered by the path obtained by NBI is a subset of the area covered by any feasible path from the source to the destination. This result generalizes to networks with eavesdroppers located in an arbitrary d-dimensional space.

Motivation for the thinnest path problem arises not only from the eavesdropper/secrecy perspective, but perhaps also from the energy efficiency perspective. Nodes that receive a signal may attempt to decode it, even if they are not in the optimal relay path. This may be particularly important in a duty-cycled sensor network where inadvertent signals may wake up sensors and cause unnecessary energy consumption.

B. Related work

There is a large body of literature on security issues in wireless ad hoc networks (see, for example, [2], [3]). However, to our best knowledge, the thinnest path problem has not been studied in the literature.

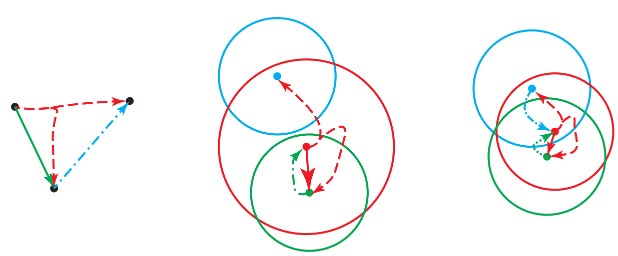
In the general context of algorithmic studies in hypergraphs, the most related work is on the shortest path problem in hypergraphs. The shortest path problem in hypergraphs remains a polynomial-time problem. The static shortest hyperpath problem was considered by Knuth [4] and Gallo *et al.* [5] in which Dijkstra's algorithm for graph was extended to obtain the shortest hyperpaths. Ausiello *et al.* proposed a dynamic shortest hyperpath algorithm for directed hypergraphs, considering only the incremental problem (i.e., network changes contain only edge insertion and weight decrease) with the weights of all hyperedges limited to a finite set of numbers [6]. In [7], Gao *et al.* developed the first fully dynamic shortest path algorithms for general hypergraphs. As discussed in I-A, the thinnest path problem studied in this paper is fundamentally different and significantly more complex than the shortest path problem.

II. Problem Formulation

A. A Directed Hypergraph Model of the Network

Consider a network with n nodes located in a d-dimensional Euclidean space. Each node v_i can choose any transmission power in the interval $[p_i, P_i]$ with $0 \leq p_i \leq P_i < \infty$. The network can be modelled by a directed hypergraph $H = (V, E)$ where V is the vertex set consisting of the n nodes and E is the set of directed hyperedges. Each directed hyperedge $e \in E$ consists of a single source vertex $s_e \in V$ and a non-empty set of destination vertices $T_e \subseteq V$ which is a distinct set of vertices that can hear the transmission of s_e for a certain feasible transmission power in $[p_{s_e}, P_{s_e}]$.

The above model allows for general non-isomorphic signal propagation in space. Isomorphic signal propagation leads to a *disk hypergraph* in which each vertex v_i is associated with a maximum[1] transmission range R_i. There exists a hyperedge $e \in E$ from a source vertex s_e to destination set T_e if and only if there exists $0 \leq r \leq R_{s_e}$ such that those and only those vertices in T_e are in the d-dimensional sphere centered at s_e with radius r. When all vertices have the same maximum transmission range $R_i = 1$, we have a *unit disk hypergraph*. Fig. 1 gives examples of directed hypergraph, disk hypergraph, and unit disk hypergraph. When all vertices have a fixed transmission power, the directed hypergraph model degenerates to a directed graph. We can similarly define disk graph (for isomorphic propagation model) and unit disk graph (for isomorphic propagation model and identical transmission power across nodes).



(a) (b) (c)

Fig. 1. (a): Directed hypergraph (b): Disk hypergraph (c): Unit disk hypergraph

B. The Width of A Hyperpath

Given a source-destination pair (s, t), a hyperpath from s to t is defined as a sequence of hyperedges $L = \{e_1, \ldots, e_m\}$ such that $s_{e_i} \in T_{e_{i-1}}$ for $1 < i \leq m$, $s_{e_1} = s$ and $t \in T_{e_m}$. The width $W(L)$ of a directed hyperpath $L = \{e_1, \ldots, e_m\}$ is defined as the total number of vertices included in all the hyperedges in L. More specifically,

$$W(L) = |\{s_{e_1} \cup T_{e_1} \cup T_{e_2} \cup \ldots \cup T_{e_m}\}|. \quad (1)$$

The thinnest path problem asks for the hyperpath from s to t with the minimum width. Note that choosing a hyperedge $e = \{s_e, T_e\}$ simultaneously chooses the relaying node s_e and its transmission power (determined by T_e).

III. The Thinnest Path Problem in 2-D Networks

In this section, we first establish the NP-completeness of the thinnest path problem in 2-dimensional networks. This result is obtained by reducing the thinnest path problem to a classic NP-complete problem—the minimum dominating set problem in graphs. This result immediately implies the NP-completeness of the thinnest path problem in higher dimensions. We then propose a polynomial-time approximation algorithm that offers a $\sqrt{\frac{n}{2}}$ approximation ratio for general directed hypergraphs and constant approximation ratios for disk hypergraphs.

A. NP-completeness

The following two theorems establish the NP-completeness of the general thinnest path problem in 2-D networks with power control.

Theorem 1: The thinnest path problem in 2-dimensional unit disk hypergraphs is NP-complete.

Theorem 2: The thinnest path problem in 2-dimensional disk graphs is NP-complete.

Note that these two theorems state stronger results than the NP-completeness of a general thinnest path problem: Theorem 1 states that the problem is NP-complete even under the isomorphic (disk) propagation model and identical maximum transmission ranges ($R_i = 1$) across nodes; Theorem 2 states that the problem is NP-complete even when each node has a fixed transmission range with isomorphic propagation model (i.e., the graph version of the problem). We point out that Theorem 1 does not imply Theorem 2 or *vice versa*. Theorem 1 pertains to networks with the freedom of power control but constrained in terms of the identical maximum transmission power across the network. Theorem 2 pertains to networks without power control but with diverse transmission powers across nodes. The NP-completeness of the thinnest path problem in one type of networks does not imply the NP-completeness in the other type of networks.

[1]We can also allow a positive minimum transmission range $r_i > 0$ and refer to the resulting hypergraph as a lower-bounded disk hypergraph.

The following theorem shows that the thinnest path problem in 3-dimensional unit disk graphs (i.e., networks with isomorphic propagation and identical and fixed transmission power at each node) is NP-complete. Whether the thinnest path problem in 2-D unit disk graphs is NP-complete is still open.

Theorem 3: The thinnest path problem in 3-dimensional unit disk graphs is NP-complete.

The proofs of these three theorems are based on reducing the thinnest path problem to the minimum dominating set problem in graphs. Recall that the minimum dominating set problem in a graph $G = \{V, E\}$ asks for the smallest subset U of V such that each vertex is either in U or a direct neighbor of a vertex in U. The minimum dominating set problem has been shown to be NP-complete with the best approximation ratio given by $\log n$. The basic idea of the proof is to show that the minimum dominating set in an arbitrary graph G leads to the thinnest path in a 2-dimensional unit disk hyergraph (for Theorem 1, and a 2-dimensional disk graph and a 3-dimensional unit disk graph for Theorems 2 and 3, respectively) specifically constructed from G. One of the main difficulties in the proofs is to maintain the geometric properties of the (unit) disk hypergraphs/graphs induced from an arbitrary graph G. In an arbitrary graph G, the neighbor set of each vertex can be defined arbitrarily while in a (unit) disk hypergraph/graph, the neighbor set of a vertex is defined based on geometric relations among vertices which would impose constraints on the composition of neighbor sets. For instance, in a 2-D unit disk graph, we can embed at most 5 neighbors to a vertex without inducing any edge among the neighboring nodes. The detailed proofs are omitted.

B. Polynomial-Time Approximation Algorithm

We propose a shortest-path based approximation algorithm for the thinnest path problem in a general directed hypergraph $H = \{V, E\}$. Specifically, for each directed hyperedge $e = \{s_e, T_e\}$, we assign weight $w(e) = |T_e|$. We then find the shortest path under this weight definition as an approximate thinnest path. Note that the shortest path in hypergraphs can be obtained in polynomial time, and efficient algorithms can be found in [5], [7]. The theorem below establishes the approximation ratio of this algorithm. The proof is omitted.

Theorem 4: The shortest-path based approximation algorithm provides an approximation ration of $\sqrt{\frac{n}{2}}$ for general directed hypergraphs, of $2(1+2\alpha)^d$ for $d-$dimensional lower bounded disk hypergraphs with $\alpha = \frac{\max_{v_i \in V} R_i}{\min_{v_i \in V} r_i}$, and of 19 for 2-dimensional unit disk graphs.

IV. The Thinnest Path Problem in 1-D and 1.5-D Networks

In this section, we consider the thinnest path problem in 1-D networks. We show that the problem is P by constructing an algorithm with time complexity of $O(n)$. Since the input data has size $O(n)$, the proposed algorithm is order-optimal. We then consider the 1.5-dimensional problem. We show that the algorithm developed for 1-D networks directly applies to the 1.5-D problem: no algorithm, even with the complete knowledge of the locations of the eavesdroppers, can obtain a thinner path than the proposed algorithm which does not require the location knowledge of the eavesdroppers.

A. 1-D Networks

Consider a network with n nodes located on a straight line. Each vertex v_i is associated with a maximum transmission range R_i and a coordinate x_i on the line (see Fig 2). Without loss of generality, we assume that $x_1 \leq x_2 \leq \ldots \leq x_n$. Let $A_{v,r}$ denote a closed d-dimensional disk centered at v with radius r. Then, we define the *cover* $A(L)$ of a hyperpath $L = \{e_1, \ldots, e_m\}$ as

$$A(L) \triangleq \bigcup_{i=1}^{m} A_{s_{e_i}, r_{e_i}} \quad (2)$$

where r_{e_i} is the minimum transmission range that induces hyperedge e_i, i.e.,

$$r_{e_i} = \min_{v \in T_{e_i}} \{d(s_{e_i}, v)\} \quad (3)$$

in which $d(\cdot, \cdot)$ is the Euclidean distance function in the space.

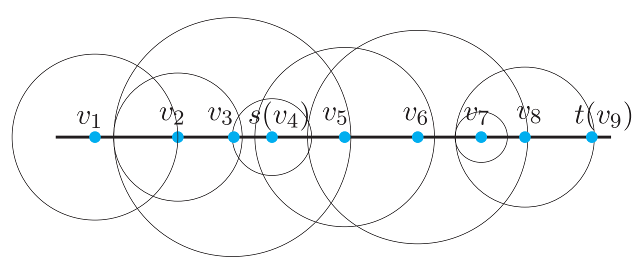


Fig. 2. A 1-D network.

1) The Algorithm based on Nested Backward Induction (NBI): It is clear that every node located between the source s and the destination t (see Fig 2) will hear the message no matter which path is chosen and all nodes to the right of t can be excluded from the thinnest path. Therefore, finding the thinnest path is to minimize the number of vertices to the left of s that can overhear the message. The problem is nontrivial since a forward path (i.e., every hop moves the message to the right toward t) from s to t may not exist and nodes to the left of s may need to act as relays. The question is thus how to efficiently find out whether a forward path exists and if not, which set of nodes to the left of s need to relay the message.

We propose an algorithm based on nested backward induction. For each vertex v, we define its predecessor ρ_v to be the rightmost vertex on the left side of v that can reach v:

$$\rho_v = \arg\max_{u \in V} \{x_u : x_u < x_v, |x_u - x_v| \leq R_u\}. \quad (4)$$

Then, in order to reach v, its predecessor ρ_v or a vertex to the left of ρ_v has to transmit. In other words, those vertices between ρ_v and v cannot reach v.

The algorithm, referred to as Nested Backward Induction (NBI), is then carried out in two steps. In the first step, the predecessors of certain vertices are obtained one by one starting from t moving toward s. Specifically, the predecessor of t, denoted by $u_1 = \rho_t$, is first obtained. If $x_{u_1} \leq x_s$, then the first step is done. Otherwise, the predecessors of u_1, denoted by $u_2 = \rho_{u_1}$, is obtained and its location compared with x_s. The same procedure continues until the currently obtained predecessor jumps to the left of s or is s itself. The first step thus produces a sequence of vertices $u_1, u_2, \ldots, u_l$ with $u_1 = \rho_t, u_2 = \rho_{u_1}$, $u_3 = \rho_{u_2}$, $\ldots$, $u_l = \rho_{u_{l-1}}$ and $x_{u_l} \leq x_s$. Let $L_1 = \{u_l, u_{l-1}, \ldots, u_1, t\}$, which is a valid path from u_l to t. If $u_l = s$, the algorithm terminates, and the thinnest path from s to t is given by L_1. Otherwise, we carry out Step 2 of the algorithm which we find a path from s to u_l. Specifically, let V' denote the set of vertices located between u_l and u_{l-1} including u_l but not u_{l-1}. The algorithm finds an arbitrary path L_2 from s to u_l that consists of only vertices in V'. Then the thinnest path L^* goes from s to u_l through L_2 and then to t through L_1.

2) The Correctness and Time Complexity of NBI: The following theorem establishes the correctness of the proposed NBI algorithm. It reveals a strong property of the path obtained by NBI: the cover of the path obtained by NBI is a subset of the cover of any feasible path from s to t. The proof is omitted.

Theorem 5: Let L^* be the path obtained by the NBI algorithm. Given any valid path L from s to t, we have $A_{L^*} \subseteq A_L$.

Theorem 6 below shows the $O(n)$ time complexity of NBI. Since the input data has size $O(n)$, NBI is order-optimal.

Theorem 6: The time complexity of the NBI algorithm is $O(n)$.

The $O(n)$ complexity of the first step of NBI can be easily established. To achieve $O(n)$ complexity in the second step, however, special care needs to be given to the specific implementation. If we directly apply a general breadth-first search (BFS) on a graph G of V' where there is a directed edge from v to u if $|x_u - x_v| \leq R_v$, the complexity would be $O(n^2)$. Below, we propose an optimized BFS with time complexity $O(n)$ by exploiting the special structure of G, .

The procedure includes two pointers, $left$ and $right$, to denote the left and right vertices that has been traversed in the search. Initially, they are both set to s. A priority queue Q is also included in the procedure as a basic part of BFS. Then, when the algorithm searches the neighbors of one vertex, only those neighbors that lie outside $[x_{left}, x_{right}]$ need to be checked. Fig. 3 demonstrates the procedure: at each iteration, the algorithm extracts a vertex z from Q and enqueues those vertices outside $[x_{left}, x_{right}]$ that can be reached by z. This iteration is repeated until u_l is reached. In order to trace back u_l to s to obtain L_2, a vector $parent$ is introduced to store the search result as a tree rooted at s. Then path L_2 is given by the path from s to u_l in this tree.

B. 1.5-D Networks

We now consider the 1.5-dimensional problem where in-network nodes are located on a line and eavesdroppers are located in a d-dimensional space that contains the line network (see Fig. 4 where eavesdroppers are located in a 2-D space).

Without loss of generality, we assume that a unit cost is incurred for each in-network node that hears the message and a non-negative cost c is incurred for each eavesdropper that hears the message. The objective is to find a path L^* from s to t with the minimum total cost:

$$L^* \stackrel{\Delta}{=} \arg \min_{L=\{e_1,\ldots,e_m\}} \{ \sum_{v \in A(L)} c(v) \} \quad (5)$$

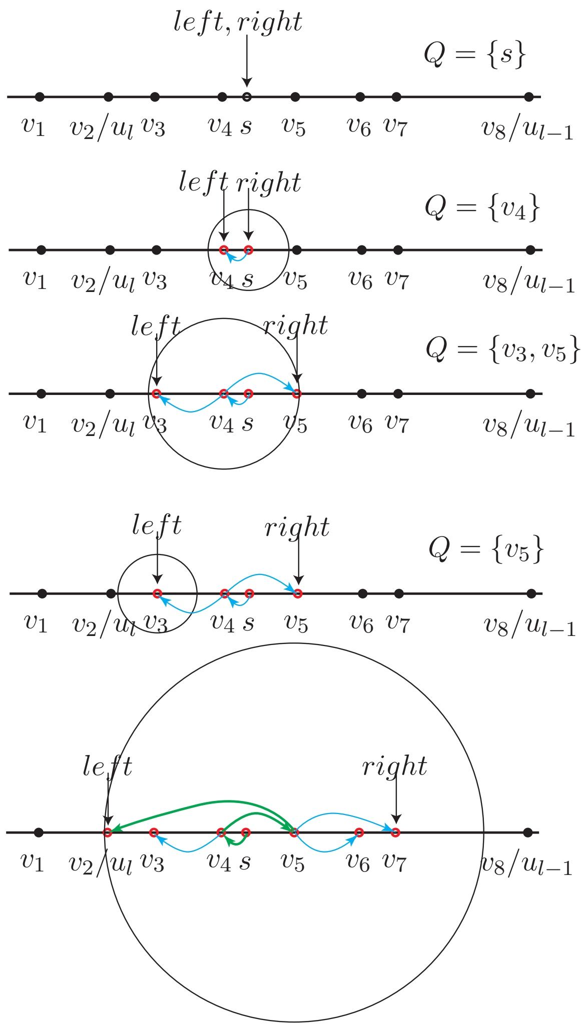


Fig. 3. An example of the optimized BFS in Step 2 of NBI (the red vertices are those between (includes) $left$ and $right$; an arrow from v to u indicates that $p[u] = v$; the procedure ends when u_l is hollow or $x_{u_l} \geq x_{left}$ and the bold arrows represent the path L_2 from s to u_l.

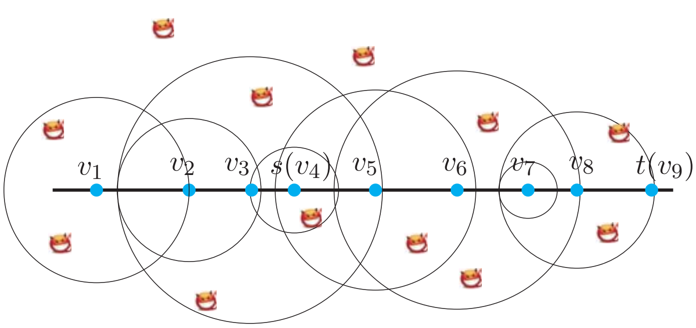


Fig. 4. An example of the thinnest path problem in the presence of eavesdroppers (the blue vertices represent in-network nodes and the devil heads the eavesdroppers).

where $c(v)$ is the cost for vertex v, and $A(L)$ is the cover of path L as defined in (2).

Based on the subset property of the cover of the path obtained by NBI as established in Theorem 5, it is easy to see that NBI provides the optimal solution to the 1.5-dimension thinnest path problem without the knowledge of the eavesdroppers locations. More specifically, no algorithm, even with the complete knowledge of the locations of the eavesdroppers, can obtain a thinner path than NBI which does not require the location knowledge of the eavesdroppers.

V. Conclusion

We have presented, to our best knowledge, the first study of the thinnest path problem. We have shown that the thinnest path problems in two-dimensional networks is NP-complete and strongly inapproximable in general. We have developed a linear-complexity algorithm based on nested backward induction for finding the thinnest path in one-dimensional networks as well as one-dimensional networks in the presence of d-dimensional eavesdroppers.

References

[1] C. Berge, *Graphs and hypergraphs*. North-Holland Pub. Co., 1976.

[2] F. Anjum and P. Mouchtaris, "Security for Wireless Ad Hoc Networks," Wiley, 2007.

[3] Y. Hu, A. Perrig, "*A Survey of Secure Wireless Ad Hoc Routing*,", Ieee Securirty and Privacy Magazing, 2(3):28-39, June 2004.

[4] D. E. Knuth, "*A generalization of dijkstra's algorithm*," Information Processing Letters, vol. 6, no. 1, pp. 177–201, February 1977.

[5] G. Gallo, G. Longo, S. Nguyen, and S. Pallottino, "*Directed hypergraphs and applications*," Discrete Applied Mathematics, vol. 42, no. 2-3, pp. 177–201, April 1993.

[6] G. Ausiello, U. Nanni, and G.F. Italiano, "Dynamic maintenance of directed hypergraphs," *Theoretical Computer Science*, vol. 72, no. 2-3, pp. 97–117, 1990.

[7] J. Gao, Q. Zhao, W. Ren, A. Swami, R. Ramanathan, A. Bar-Noy "*Dynamic Shortest Path Algorithms for Hypergraphs*," in *Proc. of the 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, May, 2012.